

Software/Tutorials available through the [Support site](#) .

- **xDEVS-core**: Discrete Event M&S kernel in Java developed in collaboration with Complutense University of Madrid, Spain. Released under GNU Lesser General Public License (LGPL)
- **DEVSML Eclipse Editor**: Eclipse DEVSML Development Studio released under Eclipse Public License (EPL). It features an auto-generated statechart and component-diagram Plantuml visualization. The execution of a coupled model generates a simulation trace that can be viewed as a Sequence diagram in the Plantuml view.

Editor Snapshots

Plug-in Development - FdsSamples/src/atomics.fds - Eclipse

File Edit Navigate Search Project Run Window Help

Package Explorer: JRE System Library [JavaSE-7], Plug-in Dependencies, src (atomics.fds, entities.fds, gpt.fds), src-gen (ent, gpAtoms, gpCoupled, gpt.java, xml.ent, xml.gpAtoms, xml.gpCoupled), META-INF, build.properties

Outline: atomics, gpAtoms, ent.*, Genr (Variables: double arrivTime, int count; Inputs: Start :: st, Stop :: sp; Outputs: Job :: job; State-Time-Advance: passive: Inf, active: arrivTime, finishing: 0.0; DEVS State Machine: start in: active; Internal-Transitions: active -> acti, finishing -> p; External-Transitions)

atomics.fds

```
31 "job.setId(count);"  
32 }  
33 deltext(S:active, X:[sp]) => S"?fini  
34 deltext(S:finishing) => S"? passive  
35 }  
36 }  
37  
38 atomic Proc{  
39     vars{  
40         double procTime  
41         Job job  
42     }  
43     interfaceIO{  
44         input Job jb  
45         output Job rs  
46     }  
47     state-time-advance{  
48         passive infinity  
49         busy procTime  
50     }  
51     state-machine{  
52         start in passive {  
53             "procTime = 3.0;"  
54         }  
55         deltext ( S: passive , X: [jb]) => S  
56             "job = jb;  
57         }  
58     }  
59     outfn (S:busy) => Y: [rs]{  
60         "rs.setId(job.getId());  
61         rs.setSolvedAt(clock);  
62     }  
63     }  
64     deltext (S:busy) => S"?passive  
65 }  
66 }  
67  
68 atomic Transd{  
69     vars{  
70         int arrivedCnt  
71         int solvedCnt  
72         double observeTime  
73         int jobId  
74     }  
75     interfaceIO{
```

PlantUML

GENR

Genr.active
ta = arrivTime

Genr.finishing
ta = 0.0

Genr.passive
ta = INF

Transd

Proc.passive
ta = INF

Problems: <terminated> gpt [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Oct 15, 2015, 7:11:34 AM)
[FINER-main|00:00:01.127]: 100.0 GpCoupled.gpt.t STATE:done,SIGMA:0.0

Snapz Pro X

Writable Insert

DEVSMML Statechart editor

Package Explorer

- JRE System Library [JavaSE-1.7]
- Plug-in Dependencies
- src
 - atomsics.fds
 - entities.fds
 - gpt.fds
- src-gen
 - ent
 - gpAtomsics
 - gpCoupled
 - gpt.java
 - xml.ent
 - xml.gpAtomsics
 - xml.gpCoupled
- META-INF
- build.properties
- logger.log

Outline

- gpt
 - gpCoupled
 - ent.*
 - gpAtomsics.*
 - gpt
 - Components
 - Genr :: g
 - Proc :: p
 - Transd :: t
 - Inputs
 - Start :: stc
 - Stop :: spc
 - Outputs
 - Result :: rsc
 - Couplings
 - EIC
 - this.stc -> g.st
 - this.spc -> g.sp
 - IC
 - g.job -> p.jb
 - g.job -> t.arrived
 - p.rs -> t.solved
 - t.ston -> g.cn

gpt.fds

```

1 package gpCoupled{
2
3
4   import ent.*
5   import gpAtomsics.*
6
7   coupled gpt{
8     models{
9       atomic Genr g
10      atomic Proc p
11      atomic Transd t
12    }
13    interfaceIO{
14      input Start stc
15      input Stop spc
16      output Result rsc
17    }
18    couplings{
19      eic this : stc -> g : st
20      eic this : spc -> g : sp
21      ic g : job -> p : jb
22      ic g : job -> t : arrived
23      ic p : rs -> t : solved
24      ic t : stop -> g : sp
25      eoc t : res -> this : rsc
26    }
27  }
28 }
  
```

PlantUML

```

graph TD
    Start((«Start»  
gpt.input.stc)) --> StartG((«Start»  
g.input.st))
    Stop((«Stop»  
gpt.input.spc)) --> StopG((«Stop»  
g.input.sp))
    StartG --> Proc[«Proc»  
p]
    StopG --> Proc
    Proc --> JobP((«Job»  
p.input.jb))
    JobP --> JobT((«Job»  
t.input.so))
    JobT --> Result((«Result»  
rsc))
  
```

Problems

<terminated> gpt [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Oct 15, 2015, 7:11:34 AM)

[FINER-main 00:00:01.127]: 100.0	GpCoupled.gpt.t	STATE:done,SIGMA:0.0
[FINEST-main 00:00:01.128]: 100.0	GpCoupled.gpt.t	OUTPUT:Result{(thruput:0.3300)(arrived:100)}
[FINEST-main 00:00:01.128]: 100.0	GpCoupled.gpt.t	OUTPUT:Stop{}
[FINEST-main 00:00:01.128]: 100.0	GpCoupled.gpt.g	INPUT:Stop{}
[FINER-main 00:00:01.128]: 100.0	GpCoupled.gpt.g	STATE:finishing,SIGMA:0.0
[FINER-main 00:00:01.128]: 100.0	GpCoupled.gpt.t	STATE:passive,SIGMA:Infinity
[FINER-main 00:00:01.129]: 100.0	GpCoupled.gpt.g	STATE:passive,SIGMA:Infinity
[FINEST-main 00:00:01.129]: 103.0	GpCoupled.gpt.p	OUTPUT:Job{(id:99)(solvedAt:100.0000)}
[FINER-main 00:00:01.129]: 103.0	GpCoupled.gpt.p	STATE:passive,SIGMA:Infinity
[FINER-main 00:00:01.129]: 103.0	GpCoupled.gpt.p	STATE:passive,SIGMA:Infinity
[FINE-main 00:00:01.129]: END SIMULATION		

BEV-SM-001: Coupling plug blue represents internal coupling and red couplings reflect

The screenshot displays the Eclipse IDE workspace for a Java project named 'FdsSamples'. The Package Explorer on the left shows the project structure, including the 'src' and 'src-gen' directories. The main editor shows the 'gpt.java' file, which defines a 'GpCoupled' class extending 'Coupled'. The class includes static variables for ports and a main method for simulation. The Console window at the bottom shows the execution output of the simulation, including timestamps and state changes for various components. The PlantUML window on the right shows the auto-generated sequence diagram for the 'GpCoupled.gpt.g' component, illustrating the interactions between different parts of the model.

```
1 package gpCoupled;
2 import gpAtoms.Genr;
3
4 public class gpt extends Coupled{
5
6     private static final Logger logger = Logger.getLogger(gpt.class);
7     public InPort<Start> stc = new InPort<>("stc");
8     public InPort<Stop> spc = new InPort<>("spc");
9     public OutPort<Result> rsc = new OutPort<>("rsc");
10
11     public static void main(String... args){
12         DevsLogger.setup(Level.FINEST); //Levels available INFO -> FINE
13         TraceCoordinator c = new TraceCoordinator(new gpt()); //with
14         //Coordinator c = new Coordinator(new gpt()); //fast-mode with
15         c.initialize();
16         c.simulate(Long.MAX_VALUE);
17     }
18
19     public gpt(){
20         this("gpCoupled.gpt", false);
21     }
22
23     public gpt(boolean debug){
24         this("gpCoupled.gpt", debug);
25     }
26 }
```

Console Output:

```
<terminated> gpt [Java Application] C:\Program Files\Java\jdk1.8.0_101\bin\javaw.exe (Oct 29, 2016, 11:11)
[FINER-main|00:00:00.484]: 20.0 GpCoupled.gpt.t STATE:active,S:
[FINEST-main|00:00:00.484]: 29.0 GpCoupled.gpt.g OUTPUT:
[FINER-main|00:00:00.484]: 29.0 GpCoupled.gpt.g STATE:active,S:
[FINER-main|00:00:00.487]: 29.0 GpCoupled.gpt.p STATE:busy,S:
[FINEST-main|00:00:00.488]: 29.0 GpCoupled.gpt.t INPUT:
[FINER-main|00:00:00.488]: 29.0 GpCoupled.gpt.t STATE:active,S:
[FINEST-main|00:00:00.488]: 30.0 GpCoupled.gpt.g OUTPUT:
[FINER-main|00:00:00.488]: 30.0 GpCoupled.gpt.p STATE:busy,S:
[FINEST-main|00:00:00.488]: 30.0 GpCoupled.gpt.t INPUT:
[FINER-main|00:00:00.488]: 30.0 GpCoupled.gpt.t STATE:active,S:
[FINER-main|00:00:00.489]: 30.0 GpCoupled.gpt.t STATE:done,S:
[FINER-main|00:00:00.489]: 30.0 GpCoupled.gpt.t STATE:done,S:
[FINEST-main|00:00:00.489]: 30.0 GpCoupled.gpt.t OUTPUT:
[FINEST-main|00:00:00.489]: 30.0 GpCoupled.gpt.t OUTPUT:
[FINEST-main|00:00:00.490]: 30.0 GpCoupled.gpt.g INPUT:
[FINER-main|00:00:00.490]: 30.0 GpCoupled.gpt.g STATE:finish:
[FINER-main|00:00:00.490]: 30.0 GpCoupled.gpt.t STATE:passive:
[FINER-main|00:00:00.490]: 30.0 GpCoupled.gpt.g STATE:passive:
[FINEST-main|00:00:00.490]: 31.0 GpCoupled.gpt.p OUTPUT:
[FINER-main|00:00:00.491]: 31.0 GpCoupled.gpt.p STATE:passive:
[FINER-main|00:00:00.491]: 31.0 GpCoupled.gpt.t STATE:passive:
[FINER-main|00:00:00.491]: END SIMULATION
```

PlantUML Diagram:

```
sequenceDiagram
    participant GpCoupled.gpt.g
    Note over GpCoupled.gpt.g: 1 [port:job] <Job:J
    Note over GpCoupled.gpt.g: 2 [port:job] <Job:J
    Note over GpCoupled.gpt.g: active
    Note over GpCoupled.gpt.g: 3 [port:job] <Job:J
    Note over GpCoupled.gpt.g: 4 [port:job] <Job:J
    Note over GpCoupled.gpt.g: active
    Note over GpCoupled.gpt.g: 5 [port:job] <Job:J
    Note over GpCoupled.gpt.g: 6 [port:job] <Job:J
    Note over GpCoupled.gpt.g: active
    Note over GpCoupled.gpt.g: 7 [port:job] <Job:J
    Note over GpCoupled.gpt.g: 8 [port:job] <Job:J
```

Simulation of a coupled model execution and auto-generated Sequence Diagram from